

## TITLE OF THE INVENTION

### IMAGE PROCESSING APPARATUS AND METHOD

5

## FIELD OF THE INVENTION

This invention relates to an image processing apparatus and method for encoding image data having object information such as attribute information.

10

## BACKGROUND OF THE INVENTION

In general, image information possesses object information, which indicates attributes, etc., of the image, as bit information, and the object information is used as important information when this image information is developed and rendered.

When image information is encoded, it is desired that the information be encoded in conformance with the image characteristics specified by attribute information, etc., possessed by this image information. When image information is encoded, therefore, an image region having certain characteristics within the image is designated and an attempt is made to encode this region using a technique that is different from that applied to other regions. In order to designate such a

region, however, the designation is made manually or the region is fixed to one that has been predetermined.

Conventionally, in a printing apparatus capable of printing a bitmapped image the basis of an input print  
5 instruction, generating attribute information about the bitmapped image by using the print instruction and attribute information of the print instruction, compressing/decompressing the bitmapped image, and outputting the bitmapped image, a single  
10 compressing/decompression parameter is set for the overall bitmapped image.

Even if a compression/decompression parameter can be set for a part of a bitmapped image, compression/decompression parameters are only switched  
15 in accordance with a target area instruction from a user, but the attribute information of the bitmapped image which is generated by using the print instruction and the attribute information of the print instruction is not used.

20 Owing to requirements for higher image quality, higher resolution, and the like, no attempt has been made to variably set compression/decompression parameters for a predetermined portion of a bitmapped image or perform efficient compression/decompression  
25 parameter control using attribute information.

## SUMMARY OF THE INVENTION

5 An object of the present invention is to provide an image processing apparatus and method for designating an image region based upon object information possessed by image information and subjecting the designated image region to encoding different from that of other regions, whereby encoding efficiency of the overall image is improved.

10 Another object of the present invention is to provide a printing apparatus which can effectively control parameter settings associated with image processing and effectively and efficiently print images while maintaining high image quality, a control method  
15 therefor, and a computer-readable memory.

According to the present invention, the foregoing object is attained by providing an image processing apparatus comprising: discrimination means for  
20 discriminating a drawing object based upon object information corresponding to the drawing object; developing means for developing the drawing object and obtaining developed image data which represents a developed image; designation means for designating a  
25 region of the developed image based upon the object information corresponding to the drawing object that has

been developed by said developing means; shift-up means  
for shifting up the bits of image data corresponding to  
the region of the developed image: and encoding means  
for entropy encoding the developed image data, in which  
5 the bits of the image data have been shifted up by said  
shift-up means.

Further, according to the present invention, the  
foregoing object is attained by providing an image  
processing method comprising: a discrimination step of  
10 discriminating a drawing object based upon object  
information corresponding to the drawing object; an  
developing step of developing the drawing object and  
obtaining developed image data which represents  
developed image; a designation step of designating a  
15 region of the developed image based upon the object  
information corresponding to the drawing object that has  
been developed by said expansion step; shift-up step of  
shifting up the bits of image data corresponding to the  
region of the developed image: and an encoding step of  
20 entropy encoding the developed image data, in which the  
bits of the image data have been shifted up by said  
shift-up step.

In a preferred aspect of the invention, the drawing  
object may be created based on a PDL data, and the  
25 object information may include an attribute information  
(such as a type of object and color and the like) of the

drawing object.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

#### BRIEF DESCRIPTION OF THE DRAWINGS

10

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

15

Fig. 1 is a block diagram illustrating the construction of an image encoding apparatus according to an embodiment of the present invention;

Fig. 2A is a diagram for describing the general construction of a wavelet transformation unit according to this embodiment;

20

Fig. 2B is a diagram for describing the functional performance of a wavelet transformation unit according to this embodiment;

Fig. 2C is a diagram for describing subbands obtained by a transformation performed by a wavelet transformation unit according to this embodiment;

Fig. 3A is a diagram for describing transformation  
5 of a region of interest (a designated region) in an image;

Fig. 3B is a diagram for describing image data before transforming in a region of interest (a designated region) in an image;

10 Fig. 3C is a diagram for describing a bit shift of image data after transformed in a region of interest (a designated region) in an image;

Fig. 4 is a diagram for describing the operation of an entropy encoder according to this embodiment;

15 Fig. 5A is a schematic view which represents the structure of a code sequence generated and outputted by spatial scalability;

Fig. 5B is a schematic view which represents the structure of a main header generated and outputted by  
20 spatial scalability;

Fig. 5C is a schematic view which represents the structure of a tile header generated and outputted by spatial scalability;

Fig. 5D is a schematic view which represents the  
25 structure of a bit stream generated and outputted by spatial scalability;

Fig. 6 is a block diagram illustrating the construction of an image decoding apparatus according to this embodiment;

Fig. 7 is a diagram for describing bit planes and  
5 the order of decoding of the bit planes in accordance with an entropy decoder according to this embodiment;

Fig. 8A is a block diagram illustrating the general construction of a wavelet decoder according to this embodiment;

10 Fig. 8B is a block diagram illustrating the functional performance of a wavelet decoder according to this embodiment;

Fig. 9A is a diagram for describing an example of a code sequence and each subband when the code sequence is  
15 decoded in the case of spatial scalability;

Fig. 9B is a diagram for describing the sizes of images displayed in correspondence with subbands and a change in reproduced images that accompanies the decoding of the code sequence of each subband in the  
20 case of spatial scalability;

Fig. 10A is a diagram for describing an example of a structure of an overall code sequence in case of SNR scalability;

Fig. 10B is a diagram for describing an example of  
25 a main header of a code sequence in case of SNR scalability;

Fig. 10C is a diagram for describing an example of a title header of code sequence in case of SNR scalability;

Fig. 10D is a diagram for in describing the  
5 sturucture of a bit stream in case of SNR scalability;

Fig. 11A is a diagram for describing a code sequence and processing for the decoding thereof in case of SNR scalability;

Fig. 11B is a diagram showing the quality of an  
10 image generated by describing a code sequence and processing for the decoding thereof in case of SNR scalability;

Fig. 12 is a diagram for describing processing of an object to be printed in a printing apparatus  
15 according to this embodiment;

Fig. 13 is a diagram for describing the structure of 3-bit information according to this embodiment; Fig. 14 is a diagram for describing combinations of objects and a flag combining method according to this  
20 embodiment;

Fig. 15 is a diagram for describing combinations of objects and a flag combining method according to this embodiment;

Fig. 16 is a diagram for describing region  
25 designation processing based upon bit information according to this embodiment;

Fig. 17 is a flowchart for describing region designation processing in a printing apparatus according to this embodiment;

Fig. 18 shows a sectional view showing the internal structure of an LBP according to a fifth embodiment;

Fig. 19 is a block diagram showing a schematic arrangement of a control system of the LBP according to the fifth embodiment in Fig. 18; and

Figs. 20A and 20B are views showing the detailed arrangement of an attribute bitmapped image in the fifth embodiment, and Fig. 20C shows an attribute bitmapped image generated by a renderer.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will now be described in detail with reference to the accompanying drawings.

A characterizing feature according to this embodiment resides in the provision of an image processing apparatus and method for deciding a region of interest in an image information on the basis of information that accompanies this image information, e.g., information indicative of data format, type of image and whether the image is color or monochrome, and encoding this region of interest at a compression ratio

held below that of other regions, whereby a decline in the image quality of the region of interest due to the encoding thereof can be prevented while the compression ratio of the overall image information is raised.

5        Fig. 1 is a block diagram illustrating the construction of an image data encoding apparatus according to an embodiment of the present invention.

As shown in Fig. 1, the apparatus includes an image input unit 101 for inputting image data. By way of  
10        example, the image input unit 1 is equipped with a scanner for reading a document image, with an imaging device such as a digital camera, or with an interface for interfacing a communication line. The input image is applied to a discrete wavelet transformation unit  
15        102, which applies a two-dimensional discrete wavelet transform to the input image data. A quantizer 103 quantizes coefficients obtained by the discrete wavelet transformation, and an entropy encoder 104 applies entropy encoding to the coefficients quantized the  
20        quantizer 103. A code output unit 105 outputs the code obtained by encoding performed by the entropy encoder 104, and a region designation unit 106 designates a region of interest in the image that has been input by the image input unit 101.

25        The apparatus according to this embodiment is not limited to a special-purpose apparatus of the kind shown

in Fig. 1 and is applicable also to a case where a program which implements these functions is loaded in, e.g., a general-purpose personal computer or work station and the computer or work station is made to  
5 operate in accordance with the program.

In the arrangement described above, first an image signal constituting an image to be encoded is input by the image input unit 101 in the order of raster scanning. The output of the image input unit 101 is  
10 input to the discrete wavelet transformation unit 102. In the description that follows, it will be assumed that the image signal that has entered from the image input unit 101 is a monochrome multilevel image data. However, if an image signal having a plurality of color  
15 components, such as a color image, is input and encoded, it will suffice to compress the RGB color components or the luminance and chromaticity components as the monochrome components.

The discrete wavelet transformation unit 102  
20 subjects the input image signal to two-dimensional discrete wavelet transform processing, calculates the transform coefficients and outputs these coefficients.

Figs. 2A to 2C are diagrams useful in describing the basic construction and operation of the discrete  
25 wavelet transformation unit 102.

As shown in Fig. 2A, the image signal input from

the image input unit 101 is stored in a memory 201. The image signal is read out of the memory 201 successively and transformed by a processor 202, after which the transformed signal is written to the memory 201 again.

5 In this embodiment, the processor 202 implements processing as shown in Fig. 2B. Here the entered image signal is separated into signals of even and odd addresses by a combination of a delay element 204 and downsamplers 205, and filter processing is applied by  
10 two filters p and u. Characters s and d represent low-pass and high-pass coefficients, respectively, when each one-dimensional image signal has been subjected to one level of decomposition. Assume that these coefficients are calculated in accordance with the following  
15 equations:

$$d(n) = x(2n+1) - \text{floor}[\{x(2n) + x(2n+2)\}/2] \quad \dots (1)$$

$$s(n) = x(2n) + \text{floor}[\{d(n-1) + d(n)\}/4] \quad \dots (2)$$

where  $x(n)$  represents an image signal that is to be transformed. In the above equations,  $\text{floor}[X]$   
20 represents the largest integral value that does not exceed  $X$ .

By virtue of the processing set forth above, one-dimensional discrete wavelet transform processing is applied to the image signal from the image input unit  
25 101. A two-dimensional discrete wavelet transform implements a one-dimensional discrete wavelet

transformation successively in horizontal and vertical directions of an image. As the details of this processing is well known, the processing will not be described here.

5        Fig. 2C is a diagram illustrating an example of a group of transformation coefficients of two levels obtained by two-dimensional discrete wavelet transform processing. The image signal is decomposed into coefficient sequences HH1, HL1, LH1, LH2, HH2, HL2, and 10 LL of different frequency bands. These coefficient sequences shall be referred to as "subbands" in the description that follows. The output of the discrete wavelet transformation unit 102 is delivered to the quantizer 103 in units of the subbands thus obtained.

15        Within an image to be encoded, a region of interest (ROI) that is to be decoded at an image quality higher than that of its surroundings is decided by the region designation unit 106, and mask information indicating which coefficients belong to the designated region is 20 generated by the region designation unit 106 when the image of interest is subjected to the discrete wavelet transform.

Fig. 3A illustrates an example of a case where mask information is generated.

25        If a star-shaped ROI has been designated in an image by a prescribed command input, as indicated on the

left side of Fig. 3A, the region designation unit 106 calculates the portion of each subband occupied by this ROI when the image which contains this designated region is subjected to the discrete wavelet transform. The  
5 region indicated by the mask information is a domain, which includes the transform coefficients of the surroundings, necessary when decoding the image signal on the boundary of the designated region.

An example of the mask information thus calculated  
10 in shown on the right side of Fig. 3A. In this example, mask information for when a two-level two-dimensional discrete wavelet transform is applied to the image on the left side of Fig. 3A is calculated a shown on the right side of Fig. 3A. In Fig. 3A, the star-shaped  
15 portion is the designated region, the bits constituting the mask information within this region are "1"s and the bits of the other mask information are "0"s. The entirety of this mask information is identical with the constitution of the transform coefficients obtained by  
20 the two-dimensional discrete wavelet transform. By scanning the bits within the mask information, therefore, it is possible to identify whether the coefficients at the corresponding positions fall within the designated region. The mask information thus  
25 produced is applied to the quantizer 103.

Furthermore, the region designation unit 106

receives an input of a parameter, which specifies the image quality of the designated region, from input means, not shown. The parameter may be a numerical value expressing a compression ratio assigned to the designated region, or a numerical value representing the image quality of this region. On the basis of this parameter, the region designation unit 106 calculates a bit-shift quantity B for the coefficients in the designated region and outputs this to the quantizer 103 together with the mask.

The quantizer 103 quantizes the entered coefficients by a predetermined quantization step and outputs indices corresponding to the quantized values. Quantization is carried out in accordance with the following equations:

$$q = \text{sign}(c) \cdot \text{floor} \{ \text{abs}(c) / \Delta \} \dots (3)$$

$$\text{sign}(c) = 1; \quad c \geq 0 \quad \dots (4)$$

$$\text{sign}(c) = -1; \quad c < 0 \quad \dots (5)$$

where c represents a coefficient that undergoes quantization and abs(c) indicates represents the absolute value of c. Further, in this embodiment, it is assumed that "1" is included as a value of the quantization step  $\Delta$ . When the value is "1", this is equivalent to a situation in which quantization is not carried out.

Next, the quantizer 103 changes the quantization

indices in accordance with the following equations based upon the mask and shift quantity B that have entered from the region designation unit 106:

$$q' = q \times 2^B; \quad m = 1 \quad \dots (6)$$

5       $q' = q \quad ; \quad m = 0 \quad \dots (7)$

where m represents the value of a mask at the position of the quantization index. By virtue of the processing described above, only a quantization index that belongs to the designated spatial region is shifted up by B bits in the region designation unit 106.

Figs. 3B and 3C illustrate a change in quantization index by such shift-up.

In Fig. 3B, three quantization indices exist in three subbands. If the value of mask information of a quantization index that has been subjected to screening is "1" and the number B of shifts is "2", then the quantization indices after the shift will be as shown in Fig. 3C.

The quantization indices that have been changed in this manner are output to the entropy encoder 104.

The entropy encoder 104 decomposes entered quantization indices into bit planes, applies binary arithmetic encoding on a per-bit-plane basis and outputs a code stream.

Fig. 4 is a diagram for describing the operation of the entropy encoder 104 according to this embodiment.

In this example, three non-zero quantization indices exist in a region within a subband having a size of  $4 \times 4$ , and the values of these indices are "+13", "-6" and "+3". The entropy encoder 104 obtains a maximum value M  
5 by scanning this region and, in accordance with the following equation, calculates a number S of bits necessary to express the maximum quantization index:

$$S = \text{ceil}[\log_2\{\text{abs}(M)\}] \quad \dots (8)$$

where  $\text{ceil}(x)$  represents the smallest integral value  
10 that is greater than x.

In Fig. 4, the maximum coefficient value is "13" and therefore the value of the bit count S is "4" and the 16 quantization indices in the sequence are processed in units of the four bit planes. First, the  
15 entropy encoder 104 applies binary arithmetic encoding to each bit of the most significant bit plane (represented by MSB in Fig. 4) and outputs the encoded bits as a bit stream. Next, the bit plane is lowered by one level and the process is repeated. This processing  
20 is repeated until the bit plane of interest reaches the least significant bit plane (represented by LSB in Fig. 4), with each bit of the bit planes being encoded and output to the code output unit 105. When the first non-zero bit is detected in the scanning of the bit planes,  
25 then the code of this quantization index undergoes entropy encoding immediately thereafter.

Figs. 5A to 5D are schematic views representing the structure of a code sequence thus generated and output.

Fig. 5A illustrates the structure of an overall code sequence, in which MH represents a main header, TH a tile header and BS a bit stream. As shown in Fig. 5B, the main header MH possesses the size of the image to be encoded (namely the numbers of pixels in the horizontal and vertical directions of the image), tile size for when the image is divided into a plurality of tiles constituting rectangular areas, number of components representing the number of each of the color components, and component information representing the size and bit precision of each component. Since an image is not divided into tiles in this embodiment, the size and image size take on identical values. In a case where the image of interest is a monochrome multilevel image, the number of components is one.

The structure of the tile header TH is shown in Fig. 5C. The tile header TH includes tile length, which contains the bit-stream length and header length of the particular tile, an encoding parameter for the tile, mask information indicating the designated region, and number of bit shifts of the coefficients that belong to this region. The encoding parameter includes the level of the discrete wavelet transform, the type of filter, etc.

Fig. 5D illustrates the structure of a bit stream in this embodiment. Bit streams are collected on a per-subband basis and are arranged in order of increasing resolution, starting from the subband for which  
5 resolution is lowest. Furthermore, in each subband, codes are arrayed on a per-bit-plane basis from a high-order bit plane (S-1) to a low-order bit plane.

By thus arraying the codes, it is possible to perform hierarchical decoding shown in Figs. 9A and 9B,  
10 described later.

In the embodiment described above, the compression ratio of the overall image to be encoded can be controlled by changing the quantization step  $\Delta$ .

In an alternative method according to this  
15 embodiment, the low-order bits of a bit plane to be encoded in the entropy encoder 104 can also be limited (discarded) in conformity with the required compression ratio. In this case, all bit planes are not encoded; encoding is carried out from the high-order bit plane to  
20 bit planes the number of which conforms to the desired compression ratio and the result is included in the final code sequence.

If the function for limiting the low-order bit planes is utilized, only bits corresponding to the  
25 designated region shown in Figs. 3A to 3C are included in the code sequence in large quantity. In other words,

it is possible to encode only the designated area, at a low compression ratio, as an image having a high image quality.

Next, a method of decoding a bit stream that has  
5 been encoded by the image encoding apparatus set forth above will be described.

Fig. 6 is a block diagram illustrating the construction of an image decoding apparatus according to this embodiment. The apparatus includes a code input  
10 unit 601, an entropy decoder 602, an inverse quantizer 603, an inverse discrete wavelet transformation unit 604 and an image output unit 605.

The code input unit 601 inputs a code sequence that has been encoded by the above-described encoding  
15 apparatus, analyzes the header contained in this code sequence and extracts the parameter necessary for subsequent processing. If necessary, the code input unit 601 controls the flow of processing or sends the particular parameter to a succeeding processing unit.  
20 Further, the bit stream contained in the code sequence is output to the entropy decoder 602.

The entropy decoder 602 subjects the bit stream, which is output from the code input unit 601, to decoding on a per-bit-plane basis and outputs the  
25 result. The decoding procedure at such time is illustrated in Fig. 7.

Numerals 701 to 704 on the left side of Fig. 7 denote the flow through which one region of a subband to be decoded is successively decoded on a per-bit-plane basis and quantization indices are finally decoded. The  
5 bit planes are decoded in the order indicated by the arrow in Fig. 7 (701 → 702 → 703 → 704). The quantization indices that have been decoded are output to the inverse quantizer 603.

The inverse quantizer 603 decodes discrete wavelet  
10 transform coefficients from the entered quantization indices in accordance with the following equations:

$$c' = \Delta \times q/2^U ; q \neq 0 \quad \dots (9)$$

$$c' = 0 ; q = 0 \quad \dots (10)$$

$$U = B ; m = 1 \quad \dots (11)$$

15  $U = 0 ; m = 0 \quad \dots (12)$

where  $q$  represents a quantization index and  $\Delta$  a quantization step. The quantization step  $\Delta$  has a value the same as that used when encoding was performed. Further,  $B$  represents a bit-shift count read out of the  
20 tile header,  $m$  denotes the value of a mask at the position of the particular quantization index, and  $c'$  represents a decoded transformation coefficient. This is obtained by decoding a coefficient represented by  $s$  or  $d$  at the time of encoding. The transformation  
25 coefficient  $c'$  is output to the succeeding inverse discrete wavelet transformation unit 604.

Figs. 8A and 8B are block diagrams illustrating the construction and processing of the inverse discrete wavelet transformation unit 604 according to this embodiment.

5 In Fig. 8A, entered transformation coefficients are stored in a memory 801. A processing unit 802 implements a one-dimensional inverse discrete wavelet transform, reads the transformation coefficients out of the memory 801 successively and performs processing,  
10 thereby executing a two-dimensional inverse discrete wavelet transform. A two-dimensional inverse discrete wavelet transform is implemented through a procedure that is the inverse of the forward transform. The details of this are well known and need not be  
15 described. Fig. 8B shows processing blocks of the processing unit 802. Here the entered transformation coefficients are subjected to the processing of the two filters u and p. After upsampling is performed, the coefficients are superimposed and the processing block  
20 outputs an image signal x'. These processes are executed in accordance with the following equations:

$$x'(2n) = s'(n) - \text{floor}[\{d'(n-1)+d(n)\}/4] \dots (13)$$

$$x'(2n+1) = d'(n) + \text{floor}[\{x'(2n)+x'(2n+2)\}/2] \dots (14)$$

The discrete wavelet transforms in the forward and  
25 reverse directions in accordance with Equations (1), (2) and Equations (13), (14) satisfy the conditions for full

reconstruction. In this embodiment, therefore, if the quantization step  $\Delta$  is "1" and all bit planes have been decoded in the decoding of the bit planes, the decoded signal  $x'$  will agree with the signal  $x$  representing the original image.

By virtue of the above-described processing, an image is decoded and outputted to the image output unit 605. The latter may be an image display unit such as a monitor or a storage device such as a magnetic disk.

Reference is made to Figs. 9A and 9B to describe the form in which an image is displayed when an image is decoded and displayed by the procedure set forth above.

Fig. 9A is a diagram illustrating an example of code sequences, in which the basic structure is based upon Figs. 5A to 5D. Here the overall image has been set as a tile and therefore the code sequence includes a single tile header and bit stream. A bit stream BS0 includes code arrayed in order of increasing resolution starting from LL, which is the subband that corresponds to the lowest resolution. Furthermore, in each subband, codes are arrayed on a per-bit-plane basis from a high-order bit plane to a low-order bit plane.

The decoding apparatus reads in the bit stream in successive fashion and displays an image when code corresponding to each bit plane has been decoded. Fig. 9B illustrates the correspondence between each subband

and the size of a displayed image as well as a change in the image that accompanies the decoding of the code sequence within the subband. It can be understood from Fig. 9B that the code sequence corresponding to LL is  
5 read out successively and that image quality gradually improves as the decoding processing of each bit plane progresses. At this time the star-shaped portion that was the designated region at the time of encoding is reconstructed with an image quality higher than that of  
10 other portions.

The reason for this is as follows: The quantization indices that belong to the designated region of the image are bit-shifted up in the quantizer at the time of quantization. At the decoding of a bit  
15 plane, therefore, the quantization indices are decoded at an earlier time than in the other portions of the image. The decoding of the designated region at a high image quality in this manner is similar for other resolutions.

20 Furthermore, when all bit planes have been decoded, the designated region and the other portions of the image are identical in terms of image quality. However, in a case where decoding is suspended in mid-course, an image in which the designated region has been  
25 reconstructed to have an image quality higher than other portions will be obtained.

In the embodiment described above, it is possible to reduce the amount of encoded data received or processed and, as a result, control the compression ratio, by limiting (neglecting) low-order bit planes  
5 decoded in the entropy decoder 602. By adopting such an expedient, it is possible to obtain a decoded image of a desired image quality from only the encoded data of the required amount. In a case where the value of the quantization step  $\Delta$  at the time of encoding is "1" and  
10 all bit planes are decoded at the time of decoding, it is possible to realise reversible encoding/decoding in which the reconstructed image agrees with the original image.

If the function for limiting the low-order bit  
15 planes is utilised, only bits corresponding to the designated region shown in Fig. 3A are included in a quantity larger than that of other areas in the code sequence that is to be decoded. The end result, therefore, is that it is possible to obtain an effect  
20 similar to the decoding of data obtained by encoding only a designated region at a low compression ratio and as an image having a high image quality.

In the description rendered above, an example in which spatial scalability is used is illustrated. It  
25 goes without saying, however, that similar effects are obtained even in a system that uses SNR scalability.

In this case, Figs. 5A to 5D for describing the encoder are replaced by Figs. 10A to 10D, and Figs. 9A to 9D for describing the decoder are replaced by Figs. 11A to 11D. Reference will now be had to Figs. 10A to 5 11B.

Figs. 10A to 10D are schematic views representing the structure of a code sequence generated and output by SNR scalability.

Fig. 10A illustrates the structure of an overall  
10 code sequence, in which MH represents a main header, TH a tile header and BS a bit stream. As shown in Fig. 10B, the main header MH possesses the size of the image to be encoded (namely the numbers of pixels in the horizontal and vertical directions of the image), tile  
15 size for when the image is divided into a plurality of tiles constituting rectangular areas, number of components representing the number of each of the color components, and component information representing the size and bit precision of each component. Since an  
20 image is not divided into tiles in this embodiment, the tile size and image size take on identical values. In a case where the image of interest is a monochrome multilevel image, the number of components is one.

The structure of the tile header TH is shown in  
25 Fig. 10C. The tile header TH includes tile length, which contains the bit-stream length and header length

of the particular tile, an encoding parameter for the tile, mask information indicating the designated region, and number of bit shifts of the coefficients that belong to this region. The encoding parameter includes the  
5 level of the discrete wavelet transform, the type of filter, etc.

Decoding side can discriminate and decode the region of interest (ROI) of the encoding data which does not include the mask information, in a case where the  
10 above described number of bit shift is equal to the number of bits which represents each of quantization indices, namely the bits are shifted up on the condition that region of interest (ROI) and non region of interest (non-ROI) do not exist on the same bits plane.

15 Fig. 10D illustrates the structure of a bit stream in this embodiment. Bit streams are collected on a per-bit-plane basis and are arranged from a high-order bit plane (S-1) to a low-order bit plane. In each bit plane the results of encoding the bit plane of the  
20 quantization indices in each subband are arranged successively on a per-subband-basis. In Fig. 10D, S represents the number of bits necessary to express the largest quantization index. The code sequences thus generated are output to the code output unit 105.

25 By thus arraying the codes, it is possible to perform hierarchical decoding of the kind shown in Figs.

11A, described later.

Figs. 11A and 11B are diagrams useful in describing the form in which an image is displayed when the code sequence shown in Fig. 10 is decoded and displayed.

5        Fig. 11A is a diagram illustrating an example of a code sequence, in which the basic structure is based upon Figs. 10A to 10D. Here the overall image has been set as a tile and therefore the code sequence includes a single tile header and bit stream. A bit stream BS0  
10 includes code arrayed in order from the highest-order bit plane to the lowest-order bit plane.

The above-described decoding apparatus reads in the bit stream in successive fashion and displays an image when code corresponding to each bit plane has been  
15 decoded. In Fig. 11A, image quality gradually improves as the decoding processing of each bit plane progresses. The star-shaped portion that was the designated region at the time of encoding is reconstructed with an image quality higher than that of other portions.

20        As set forth earlier, the reason for this is as follows: The quantization indices that belong to the designated region of the image are bit-shifted up in the quantizer at the time of encoding. At the decoding of a bit plane, therefore, the quantization indices are  
25 decoded at an earlier time than in the other portions of the image.

Furthermore, when all bit planes have been decoded, the designated region and the other portions of the image are identical in terms of image quality. However, in a case where decoding is suspended in mid-course, an  
5 image in which the designated region has been reconstructed to have an image quality higher than other portions will be obtained.

The characterizing features of this embodiment will now be described in the case of a printing apparatus.

10 Fig. 12 is a diagram useful in describing processing of an object to be printed in a printing apparatus according to this embodiment.

In Fig. 12, a controller 1100 administers control of the overall operation of the printing apparatus according to this embodiment. The controller 1100  
15 includes a PDL analyzer 1101 for analyzing and interpreting PDL data (print data) that enters from an external device such as a host computer. A print object 1102 is created by analyzing the entered PDL data using  
20 the PDL analyzer 1101. A graphics library 1103 converts the object 1102 to a command object 1104 that can be interpreted by a rendering engine 1106 and adds information 1105 (which will be described later with reference to Fig. 13) to the object 1104. The object  
25 1104 is input to the rendering engine 1106, which proceeds to manipulate the bit information and create

bitmap image data 1107 for printing, as will be described later with reference to Figs. 14 and 15. The bitmap image data 1107 includes image data (1108) comprising R, G, B color components of eight bits each, and 3-bit information (bit/pixel) (1109).

Fig. 13 is a chart for describing the meaning of each bit of the information 1105.

As shown in Fig. 13, the information 1105 is composed of three bits. Bit 0 is a bit (referred to as a "bitmap flag") indicative of a bitmap or vector graphic. The bit is set to "0" in case of a bitmap and to "1" in case of a vector. Bit 1 is a bit (referred to as a "color flag") indicative of a color object or monochrome object. The bit is "0" in case of color and "1" in case of monochrome. Bit 2 is a bit referred to as a "character flag". In a case where the bitmap flag is "0" (bitmap data), the bit 2 is indicative of a character (= "1") or of an image other than a character (= "0"). In a case where the bitmap flag is "1" (vector data), the bit 2 indicates grayscale priority (= "0") or resolution priority (= "1").

Figs. 14 and 15 are diagrams useful in describing methods of combining a source image (SI) and a destination image (DI).

In Fig. 14, numeral 1401 indicate a case where an object to be combined is only SI (the source image only,

i.e., the object is overwritten) or only Not SI (a case where a 0/1 reversal of the source image is made and the object is overwritten). This case indicates that the 3-bit flag (SIflag) of SI is left as the 3-bit information  
5 1109. Numeral 1402 indicate a case where an object to be combined is only DI (the destination image only, i.e., the underlying background is utilized as is) or only Not DI (a 0/1 reversal of the destination bit is performed). This case indicates that the 3-bit flag  
10 (DIflag) of DI is left as the 3-bit information 1109. Numeral 1403 indicate a method of combining SI and DI. This includes "SI or DI" (the logical sum or SI and DI), "SI and DI" (the logical product of SI and DI), "SI xor DI" (the exclusive OR between SI and DI) and "SI  $\alpha$  DI"  
15 (an  $\alpha$  operation between SI and DI, namely an AND, OR or XOR operation between SI and DI having transparency). This case indicates finding the logical product between the bits of the 3-bit information of SI and the bits of the 3-bit information of DI and deciding that the result  
20 is information representing the combined image.

Fig. 15 is a diagram useful in describing another combining method. In a manner similar to 1401 above, numeral 1501 indicate a case where an object to be combined is only SI (the source image only, i.e., the  
25 object is overwritten) or only Not SI (a NOT operator of the source, meaning that a 0/1 reversal of the source

bit is performed and the object is overwritten). This case indicates that the 3-bit information of SI is left as the information 1109. In a manner similar to 1402 above, numeral 1502 indicate a case where an object to  
5 be combined is only DI (the destination image only, i.e., the underlying background is utilized as is) or only Not DI (a NOT operator of the destination, meaning that a 0/1 reversal of the destination bit is performed). This case indicates that the 3-bit  
10 information of DI is left as the information 1109. Numeral 1403 indicate a method of combining SI and DI. This includes "SI or DI" (the logical sum or SI and DI), "SI and DI" (the logical product of SI and DI), "SI xor DI" (the exclusive OR between SI and DI) and "SI  $\alpha$  DI"  
15 (an  $\alpha$  operation between SI and DI, namely an AND, OR or XOR operation between SI and DI having transparency). This case indicates that the bit information (SIflag) of SI is left as is if SI and DI are identical for each bit of the 3-bit information of SI and DI, and that 0 (000)  
20 (indicative of a bitmap, color and anything other than a character) is set as the bit information if SI and DI are not identical.

Fig. 16 is a diagram useful in describing region designation in print image data that has been generated  
25 in a printing apparatus according to this embodiment.

In Fig. 16, image data 1501 corresponds to the RGB

image data 1108 of Fig. 12, and an image 1502 is an  
image data area having an attribute 1 (e.g., 000:  
bitmap; color; anything other than a character) defined  
by the information illustrated in Fig. 13. Similarly,  
5 an image 1503 is an image data area having an attribute  
2 (e.g., 010: bitmap; monochrome; anything other than a  
character) defined by the information illustrated in  
Fig. 13. Further, an image 1504 is an image data area  
having an attribute 3 (e.g., 001: bitmap; monochrome;  
10 character) defined by the information illustrated in  
Fig. 13. Numerals 1505 denote information, equivalent  
to flag 1109 in Fig. 12, corresponding to the image data  
1501.

This embodiment is characterized in that 3-bit  
15 information is used in order to decide a region of  
interest (ROI) in the region designation unit 106 (for  
designating the ROI) of Fig. 1. More specifically, the  
information 1505 (which corresponds to the flag 1109 in  
Fig. 12) shown in Fig. 16 is entered, areas capable of  
20 being an ROI are extracted (1506) based upon this  
information, and the rendering engine 1106 decides the  
area to be used as the ROI from the areas that have been  
extracted.

The method of selecting this ROI is determined by  
25 providing a priority for the state of each bit of the 3-  
bit information shown in Fig. 13 and giving precedence

to data having an attribute which gives priority to  
image quality (an attribute for which quality will not  
decline at a high ratio of compression). For example,  
settings are made in such a manner that an image region  
5 (vector; color; resolution priority) for which the 3-bit  
information is "101" is adopted as an important high-  
quality image area, and such that an image region  
(vector; color; grayscale priority) for which the 3-bit  
information is "100" is adopted as the next most  
10 important high-quality image area.

In relation to this assignment of priority, it is  
necessary to incorporate also the print engine  
characteristics, etc., though this is not particularly  
defined here.

15 Fig. 17 is a flowchart useful in describing region  
designation processing in a printing apparatus according  
to this embodiment.

Step S1 of the flowchart calls for entry of print  
data (drawing command) from a host computer. Next, at  
20 step S2, the entered print data (drawing command) is  
analyzed and an object that can be referred to at the  
succeeding rendering processing is generated. Control  
then proceeds to step S3, at which combination of images  
and the accompanying processing for generating 3-bit  
25 information are executed, in the manner described in  
connection with Figs. 14 and 15, in accordance with a

print data command.

Control then proceeds to step S4, at which the flag 1109 is checked to determine whether the 3-bit information indicates a prescribed attribute (e.g.,  
5 vector data; color; resolution priority). If the answer is "YES", control proceeds to step S6, where the image region to which this flag is attached is set as the ROI. Next, it is determined at step S7 whether all flags have been checked. The processing of steps S5 to S7 is  
10 repeatedly executed until all flags 1109 have been checked. When a designated region has thus been decided, control proceeds to step S8. Here rendering processing is executed to generate the bitmap image data 1107 that includes the image data 1108 and flag 1109  
15 shown in Fig. 12.

The same effects are obtained through another method, namely deciding the ROI upon examining the image data 1108 of eight bits for each of R, G, B in the bitmap image data 1107 after all data have been  
20 rendered, i.e., after the bitmap image data 1107 has been obtained.

Such region designation processing is not limited to execution in a printer. It goes without saying that this processing can be applied when encoding video or a  
25 photographic image shot by an X-ray camera or digital camera. The technique for designating the region is in

no way limited to a printer.

[Second Embodiment]

In the embodiment described above, priority is assigned to bit information. However, an image having a high quality overall can be obtained by selecting as the ROI a portion for which the 3-bit information takes on the largest area and reporting this portion to the rendering engine 1106.

[Third Embodiment]

10 In the first and second embodiments, a single region is selected as the ROI. However, in a case where a plurality of regions can be designated as ROIs, a plurality of ROIs may be designated based upon the first and second embodiments.

15 It goes without saying that in the first to third embodiments, the region designation method of the present invention can be adapted to all compression schemes that have a function equivalent to an ROI.

[Fourth Embodiment]

20 The first to third embodiments set forth above illustrate an example in which an ROI is found from 3-bit information. However, in a printing apparatus that employs a plurality of compression methods that do not use an ROI, it is possible to use the bit information as a reference for selecting the compression method. As  
25 one example, consider a case where the printer supports

PackBits compression and JPEG compression. In such case the bit information is examined and JPEG compression is used for an image having many color images and PackBits compression is used for an image having many monochrome characters. Furthermore, the combination of the compression method and bit information can be decided substantially automatically by the characteristics of the compression method.

It goes without saying that the present invention may include in a case where quantization process or inverse quantization process is omitted from the encoding step and the decoding step in the above embodiments. In such cases, the coefficients prior to quantization, instead of the quantization index, can be shifted up and then each bits plane obtained by the quantization can be encoded.

[Fifth Embodiment]

A fifth embodiment in which the present invention is applied to a color laser beam printer (to be abbreviated to LBP hereinafter) will be described below. In the fifth embodiment, a color laser beam printer is described as an example of a printing apparatus. However, the present invention can be applied to other types of color printers such as a color ink-jet printer and color thermal transfer printer.

The arrangement of the LBP will be described first with reference to Fig. 18.

Fig. 18 is a sectional view showing the internal structure of the LBP according to the fifth embodiment.

5 Referring to Fig. 18, reference numeral 2100 denotes an LBP body which generates character patterns, graphic patterns, images, and the like and form them on printing media as printing sheets in accordance with character print instructions (drawing command), various  
10 graphic pattern print instructions (drawing command), image print instruction, color designation instructions, and the like supplied from a host computer (2201 in Fig. 19).

Reference numeral 2151 denotes an operation panel  
15 on which switches for operation, an LED display unit for displaying the state of the LBP body 2100, an LCD unit, and the like are arranged; and numeral 2101 denotes a printer control unit for controlling the overall LBP body 2100 and analyzing character print instructions and  
20 the like supplied from the host computer. To convert pieces of R, G, and B color information into M (Magenta), C (Cyan), Y (Yellow), and K (black), the LBP according to the fifth embodiment includes M, C, Y, and K image forming/developing mechanisms. The printer  
25 control unit 2101 generates M, C, Y, and K print images,

converts them into video signals, and outputs them to M, C, Y, and K laser drivers.

An M (Magenta) laser driver 2110 is a driving circuit for a semiconductor laser 2111, which  
5 on/off-controls a laser beam 2112 emitted from the semiconductor laser 2111 in accordance with an input video signal. The laser beam 2112 is laterally deflected by a rotating polygon mirror 2113 to scan an electrostatic drum 2114. With this operation, an  
10 electrostatic latent image such as a character or graphic pattern is formed on the electrostatic drum 2114. This electrostatic latent image is developed by a developing unit (toner cartridge) 2115 located around the electrostatic drum 2114 and then transferred onto a  
15 printing sheet.

For C (Cyan), Y (Yellow), and K (black), the LBP has image forming/developing mechanisms similar to the one for M (Magenta). Reference numerals 2120, 2121, 2122, 2123, 2124, and 2125 denote image  
20 forming/developing mechanisms for C (Cyan); 2130, 2131, 2132, 2133, 2134, and 2135, image forming/developing mechanisms for Y (Yellow); and 2140, 2141, 2142, 2143, 2144, and 2145, image forming/developing mechanisms for K (black). Each mechanism has the same function as that  
25 of the M (Magenta) image forming/developing mechanism.

Since these mechanisms have the same arrangement, a description thereof will be omitted.

As printing paper, printing cut sheets are used, which are stored in a paper feed cassette 2102 mounted  
5 in the LBP body 2100 and held at a predetermined level by a spring 2103. A cut sheet is fed into the apparatus by a feed roller 2104 and convey rollers 2105 and 2106 and conveyed on a paper convey belt 2107 to pass through each image forming/developing unit.

10 M, C, Y, and K toners (powder inks) transferred onto the printing sheet are fixed thereon by a fixing unit 2108 with heat and pressure. The printing sheet is then output to the upper portion of the LBP body 2100 by convey rollers 2109 and 2150.

15 Fig. 19 is a block diagram showing the schematic arrangement of the control system of the LBP according to the fifth embodiment in Fig. 18.

The printer control unit 2101 of this LBP receives data 2218 including character, graphic pattern, and  
20 image print instructions, color information, and the like sent from the host computer 2201 serving as a print information source. A CPU 2208 processes the received data. The resultant character information and the like are printed in units of pages.

25 Reference numeral 2202 denotes an input/output interface unit for exchanging various kinds of

information with the host computer 2201; and numeral 2203 denotes an input buffer memory for temporarily storing various kinds of information input through the input/output interface unit 2202.

5           Reference numeral 2204 denotes a character pattern generator which is comprised of a font information unit 2222 in which attributes such as the widths and heights of characters and the addresses of actual character patterns are stored, a character pattern unit 2223 in  
10   which character patterns themselves are stored, and a control program (not shown) for reads from these units. The control program is stored in a ROM 2219 and also has a code convert function for calculating the address of a character pattern corresponding to a received character  
15   code.

          Reference numeral 2205 denotes a RAM including a font cache area 2207 in which character patterns output from the character pattern generator 2204 are stored and a storage area 2206 in which external character font  
20   information, form information, current print environment, and the like sent from the host computer 2201 are stored.

          Storing pattern information, which is temporarily bitmapped into a character pattern, in the font cache  
25   area 2207 as a font cache in advance eliminates the necessity to decode the same character and bitmap it

into a pattern every time the same character is printed.  
Hence, pattern information can be quickly bitmapped into  
a character pattern.

The CPU 2208 controls the overall control system  
5 of the LBP and controls the overall apparatus according  
to the control program (not shown) for the CPU 2208  
which is stored in the ROM 2219. Reference numeral 2209  
denotes a intermediate buffer in which an internal,  
intermediate data group generated on the basis of the  
10 input data 2218 is temporarily stored.

After 1-page data are completely received,  
converted into simpler intermediate data, and stored in  
the intermediate buffer 2209, the intermediate data are  
rendered several lines at a time by a renderer 2210 and  
15 output as print image to a band buffer 2211.

Note that the renderer 2210 can generate print  
bitmapped images with 8 bits/pixel for each of R, G, and  
B several lines at a time, and at the same time, can  
generate an attribute bitmapped image with 2 bits/pixel,  
20 which indicates whether each pixel is a character,  
graphic pattern, or image pixel.

The band buffer 2211 has a storage area in which  
at least 8-line R, G, and B print bitmapped images and  
attribute bitmapped images can be stored. At this time,  
25 the print bitmapped images and attribute bitmapped  
images are separately compressed by a compression unit

2212. When the R, G, and B print bitmapped images are to be compressed, attribute bitmapped images are input as compression parameters.

The images output to the band buffer 2211 are  
5 compressed in units of scan lines by the compression unit 2212 and then stored in a page memory 2213. After rendering and compression of 1-page intermediate data are completed and the resultant data are stored in the page memory 2213, the compressed data are read out and  
10 decompressed several lines at a time by a decompression unit 2214. At this time, pint bitmapped images and attribute bitmapped images are separately read out and decompressed.

The bitmapped images with 8 bits/pixel for each of  
15 R, G, and B as the decompressed data are converted into bitmapped images with 4 bits/pixel for each of Y, M, C, and K by a color conversion unit 2215. When each pixel of the pint bitmapped images is to be color-converted, color conversion methods are switched in accordance with  
20 the corresponding type of attribute bitmapped pixel. An output interface unit 2216 having output interface units for Y, M, C, and K converts the bitmapped images with 4 bits/pixel for each of Y, M, C, and K into a video signal and outputs it to a printer unit 2217. The  
25 printer unit 2217 is the printing mechanism portion of

the page printer which prints image information based on the video signal from the output interface unit 2216.

As described above with reference to Fig. 18, in the LBP according to the fifth embodiment, to

5 concurrently perform image forming/developing for M, C, Y, and K, the output interface unit 2216 is comprised of four output interface units, namely an M output interface unit, C output interface unit, Y output interface unit, and K output interface unit. These

10 interface units independently acquire dot data from the color conversion unit 2215, converts the data into video signals, and outputs the signals to the laser drivers 2110, 2120, 2130, and 2140 in the respective planes.

Reference numeral 2220 denotes a nonvolatile

15 memory generally formed by an EEPROM or the like and will be referred to as an NVRAM (Nonvolatile RAM) hereinafter. The NVRAM 2220 is used to store the panel set values designated by the operation panel 2151. Reference numeral 2221 denotes data to be transmitted

20 from the LBP 2100 to the host computer 2201.

Note that the ROM 2215 includes control programs for analyzing data (drawing command) input from the host computer 2201, generating intermediate data, and controlling the printing mechanism body 2213, a color

25 conversion table for conversion from an RGB color space to an MCYK color space, and the like.

The attribute bitmapped images generated by the  
renderer 2210 will be described next with reference to  
Figs. 20A and 20B.

Figs. 20A and 20B are views showing the detailed  
5 arrangement of an attribute bitmapped image in this  
embodiment.

Fig. 20A shows a print bitmapped image generated  
by the renderer 2210. Fig. 20B shows intermediate data  
stored in the intermediate buffer 2209. Fig. 20C shows  
10 an attribute bitmapped image generated by the renderer  
2210.

First of all, it is checked whether an input  
drawing command is a character, graphic pattern, or  
image command, and a flag is set for each intermediate  
15 data stored in the intermediate buffer 2209 (Fig. 20B).

More specifically, the print bitmapped image in  
Fig. 20A contains a character 2100a, graphic pattern  
(circle) 2100b, and image 2100c. As intermediate data,  
Object 1 corresponding to the character 2100a contains  
20 an attribute (type) : character (char), print position  
(x, y), font name (font-name), size (size), character  
code (code = a, b, c), and color (color). Object 2  
corresponding to the graphic pattern 2100b contains an  
attribute (type) : graphic pattern (graphic), print  
25 position (x, y), radius (r), shape (circle), and color.  
Object 3 corresponding to the image 2100c contains an

attribute (type) : image (image), print position (x, y),  
image width (w), image height (h), and pointer (pointer)  
for an image substance.

As described above, the intermediate data contain  
5 pieces of information about the shapes, colors, and  
print positions of the respective print objects.

When this intermediate data is printed by the  
renderer 2210, an attribute bitmapped image (Fig. 20C)  
is obtained, together with the print bitmapped image  
10 (Fig. 20A).

With regard to the attribute bitmapped image, each  
pixel of a character information area "0x1", graphic  
pattern information area "0x2", image information area  
"0x3", and remaining area "0x0" is expressed by 2-bit  
15 information.

According to the fifth embodiment, an ROI is  
determined and compression parameters are automatically  
set on the basis of the attribute bitmapped image data  
of input data, and hence an image can be effectively and  
20 efficiently printed while good image quality is  
maintained.

The contents of the attribute bitmapped image  
generated by the renderer 2210 may be the types of  
objects, e.g., a character, graphic pattern, and image.

25 In the fifth embodiment, the contents of the  
attribute bitmapped image generated by the renderer 2210

may be a type indicating the output quality of an area, i.e., representing a high-quality area for a high-quality output or a poor-quality area for poor-quality output.

- 5           In the compression unit 2212 (encoder), as a mask area of an ROI for the encoder, the OR of the character information bits and graphic pattern information bits of an attribute bitmapped image may be used.

- 10           In the fifth embodiment, the compression unit 2212 (encoder) may use the high-quality object information of an attribute bitmapped image as a mask area of an ROI.

- The ROM 2215 stores control programs for analyzing data input from the host computer 2201, generating intermediate data, and controlling the printing  
15           mechanism body 2213, a color conversion table for conversion from an RGB color space to an MCYK color space, and the like. In the fifth embodiment, in particular, data in the form of print instructions such as character print instructions, various graphic pattern  
20           print instructions, image print instructions, color designation instructions, and the like are analyzed. Color area separation can also be added to this data analysis. By adding this image area separation, the present invention can be applied to even a case wherein  
25           bitmapped image data is input from the host computer 2201. That is, image area separation is performed for

input bitmapped image data, and an attribute bitmapped image can be determined from the bitmapped image data having undergone image area separation. Alternatively, the bitmapped image data generated by the host computer  
5 2201 may be subjected to image area separation on the host computer 2201, and the image area separation result and the bitmapped image data may be input.

As a mask area of an ROI, a predetermined attribute bitmapped image is fixed in the fifth  
10 embodiment. However, the user may select a desired attribute bitmapped image as a mask area of an ROI by using, for example, the operation panel 2151.

The present invention can be applied to a system constituted by a plurality of devices (e.g., a host computer, interface, reader, printer, etc.) or to an apparatus comprising a single device (e.g., a copier or  
5 facsimile machine, etc.).

Furthermore, it goes without saying that the object of the invention is attained also by supplying a storage medium (or recording medium) storing the program codes of the software for performing the functions of the  
10 foregoing embodiment to a system or an apparatus, reading the program codes with a computer (e.g., a CPU or MPU) of the system or apparatus from the storage medium, and then executing the program codes. In this case, the program codes read from the storage medium  
15 implement the novel functions of the embodiments and the storage medium storing the program codes constitutes the invention. Furthermore, besides the case where the aforesaid functions according to the embodiments are implemented by executing the program codes read by a  
20 computer, it goes without saying that the present invention covers a case where an operating system or the like running on the computer performs a part of or the entire process in accordance with the designation of program codes and implements the functions according to  
25 the embodiments.

It goes without saying that the present invention

further covers a case where, after the program codes read from the storage medium are written in a function expansion card inserted into the computer or in a memory provided in a function expansion unit connected to the  
5 computer, a CPU or the like contained in the function expansion card or function expansion unit performs a part of or the entire process in accordance with the designation of program codes and implements the function of the above embodiments.

10 In accordance with the embodiments as described above, a region of interest in image information having attribute information as bit information is decided using the bit information, thereby making it possible to obtain an image having a higher quality and a higher  
15 compression.

Further, by providing an attribute with a degree of priority and setting an image region that has an attribute with a high degree of priority as a region of interest, an image can be encoded without lowering the  
20 compression ratio of the overall image while suppressing a decline in the compression ratio of a region in which resolution is important.

In accordance with the present invention, as described above, an image region is designated based  
25 upon object information possessed by image information and the designated image region is subjected to encoding

that is different from that of other regions, thereby making it possible to raise the encoding efficiency of the overall image.

As many apparently widely different embodiments of  
5 the present invention can be made without departing from the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the appended claims.